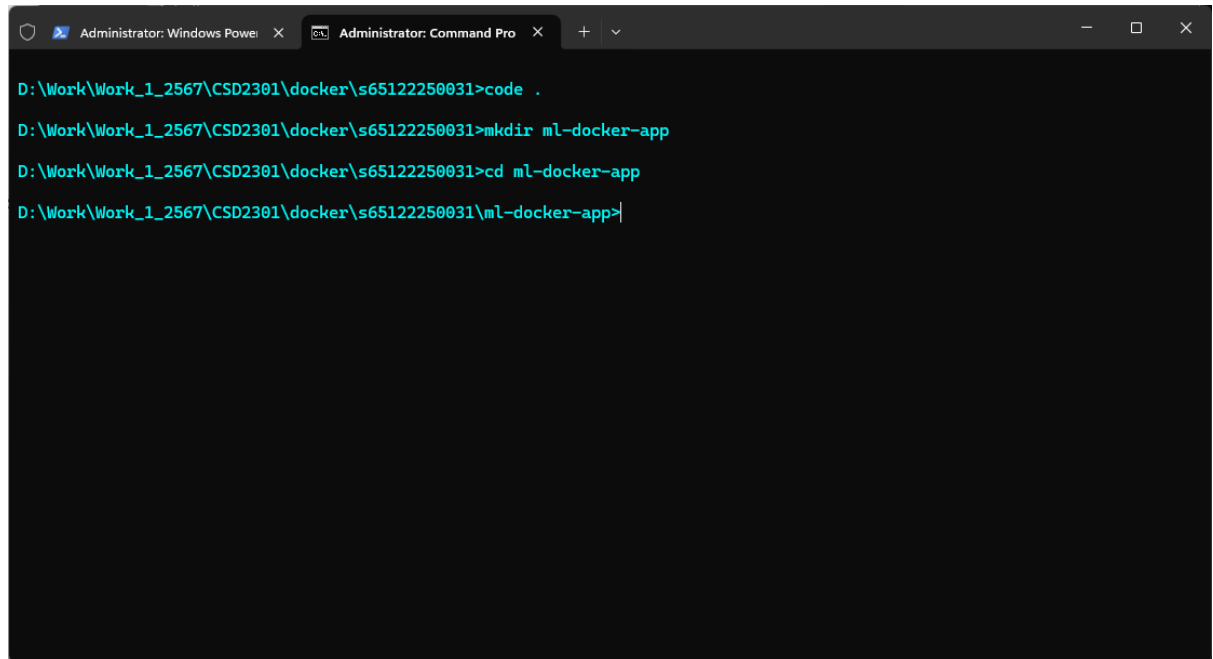


Lab ML and Docker

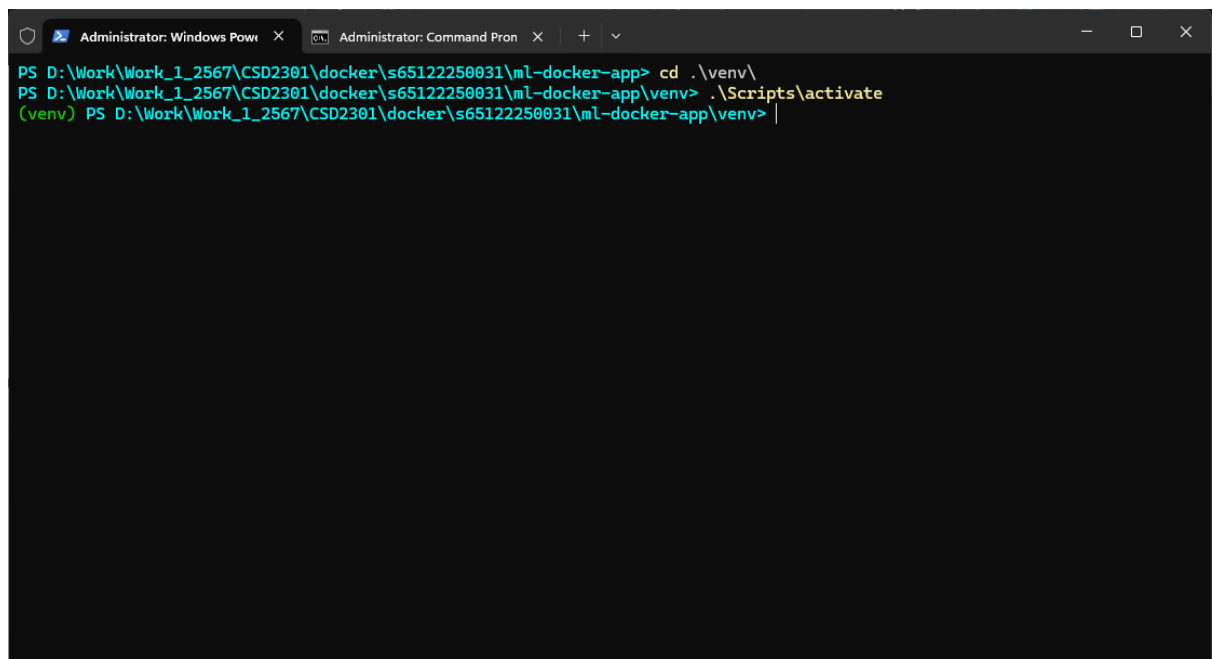
1. Docker Installed
2. Run Power Shell as Administrator
3. Project directory



A screenshot of a Windows Command Prompt window titled "Administrator: Command Prom". The window shows the following commands and their outputs:

```
D:\Work\Work_1_2567\CSD2301\docker\s65122250031>code .  
D:\Work\Work_1_2567\CSD2301\docker\s65122250031>mkdir ml-docker-app  
D:\Work\Work_1_2567\CSD2301\docker\s65122250031>cd ml-docker-app  
D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app>
```

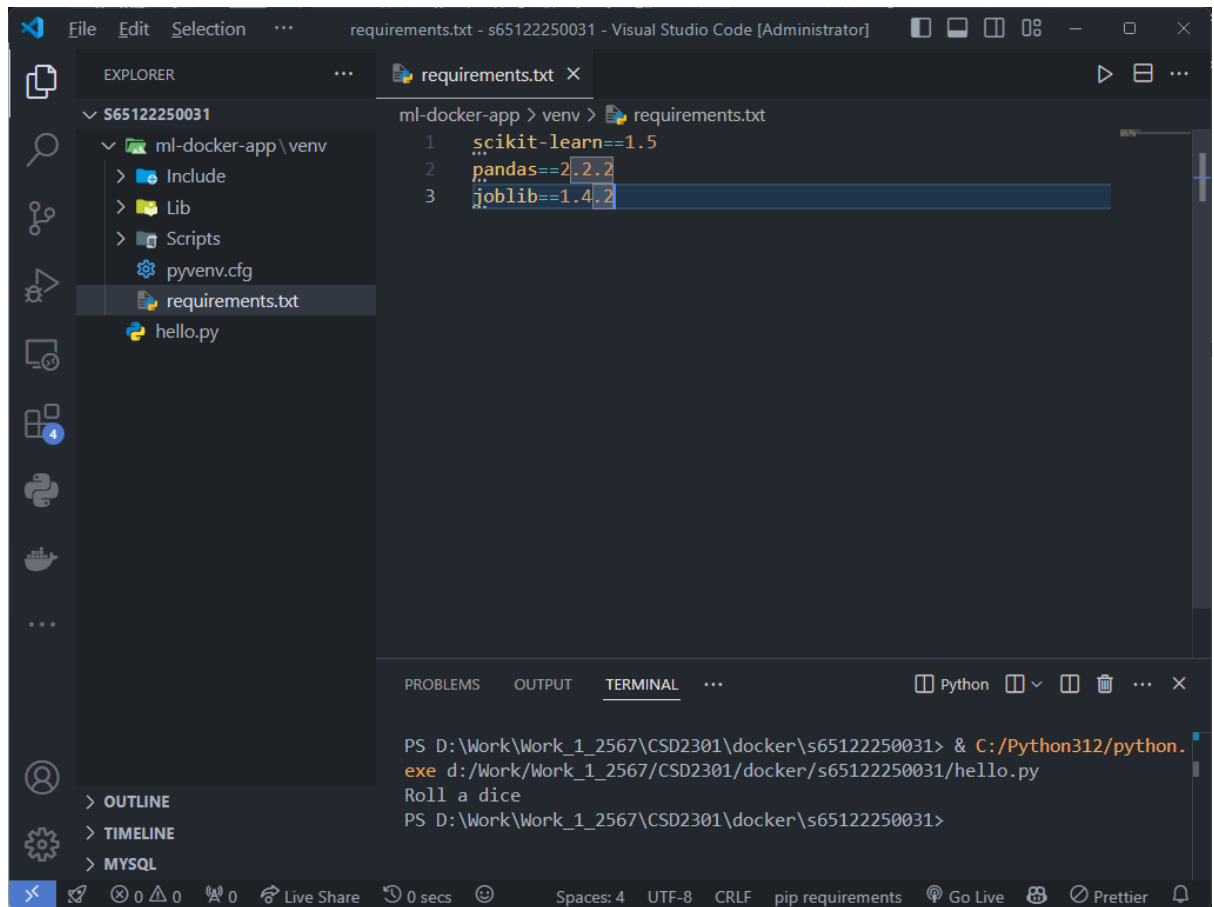
4. Set Up a Python Virtual Environment



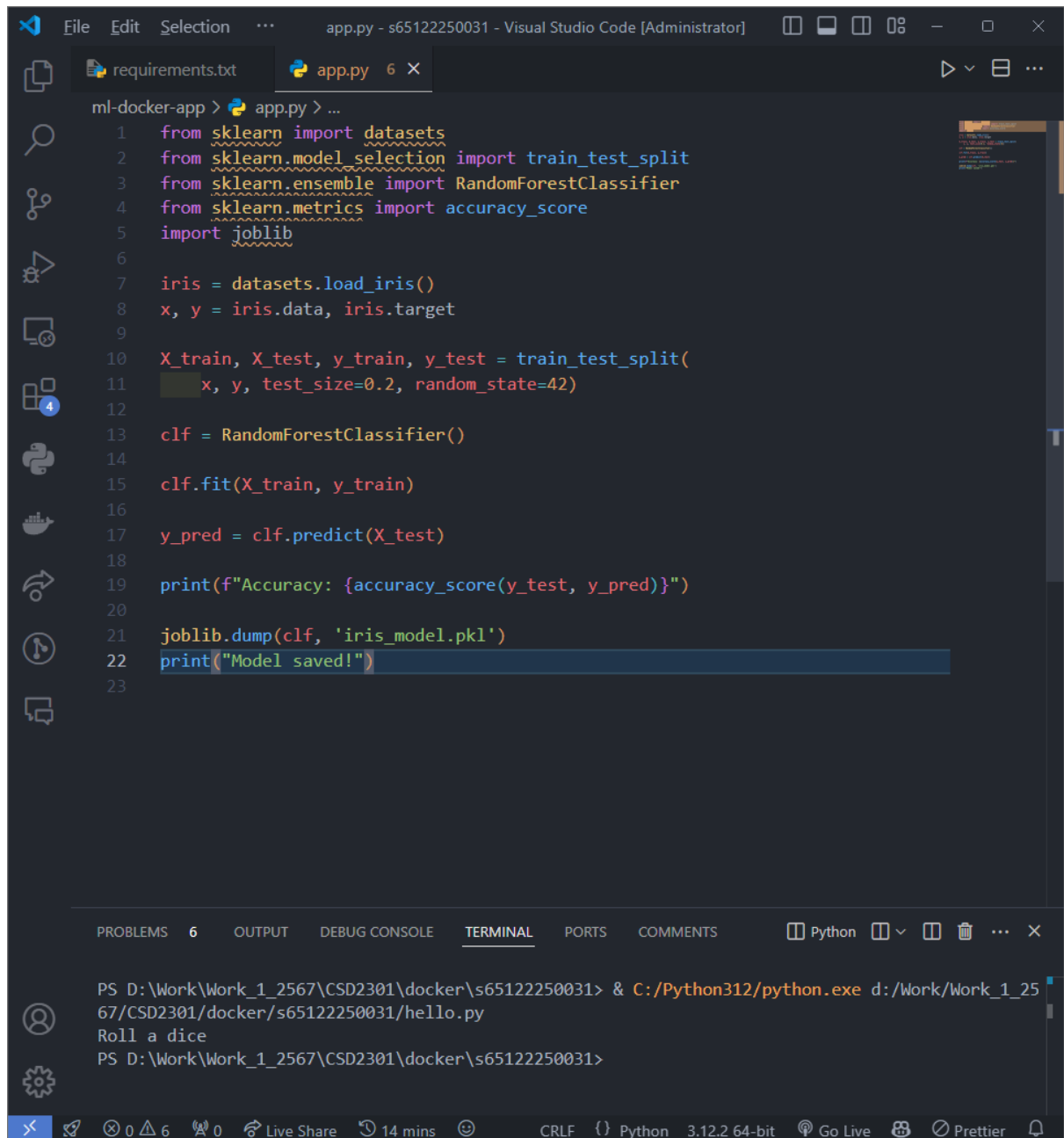
A screenshot of a Windows Command Prompt window titled "Administrator: Command Prom". The window shows the following commands and their outputs:

```
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app> cd .\env\  
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\env> .\Scripts\activate  
(env) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\env>
```

5. Requirement.txt File



6. Script



The image shows a Visual Studio Code editor window with a Python file named `app.py` open. The script is a machine learning application that uses the Iris dataset, sklearn, and joblib. The code is as follows:

```
1 from sklearn import datasets
2 from sklearn.model_selection import train_test_split
3 from sklearn.ensemble import RandomForestClassifier
4 from sklearn.metrics import accuracy_score
5 import joblib
6
7 iris = datasets.load_iris()
8 x, y = iris.data, iris.target
9
10 X_train, X_test, y_train, y_test = train_test_split(
11     x, y, test_size=0.2, random_state=42)
12
13 clf = RandomForestClassifier()
14
15 clf.fit(X_train, y_train)
16
17 y_pred = clf.predict(X_test)
18
19 print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
20
21 joblib.dump(clf, 'iris_model.pkl')
22 print("Model saved!")
23
```

Below the editor, the TERMINAL panel is active, showing the command prompt and the execution of the script:

```
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031> & C:/Python312/python.exe d:/Work/Work_1_2567/CSD2301/docker/s65122250031/hello.py
Roll a dice
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031>
```

The status bar at the bottom indicates the file is a Python script (3.12.2 64-bit) and shows various icons for debugging and formatting.

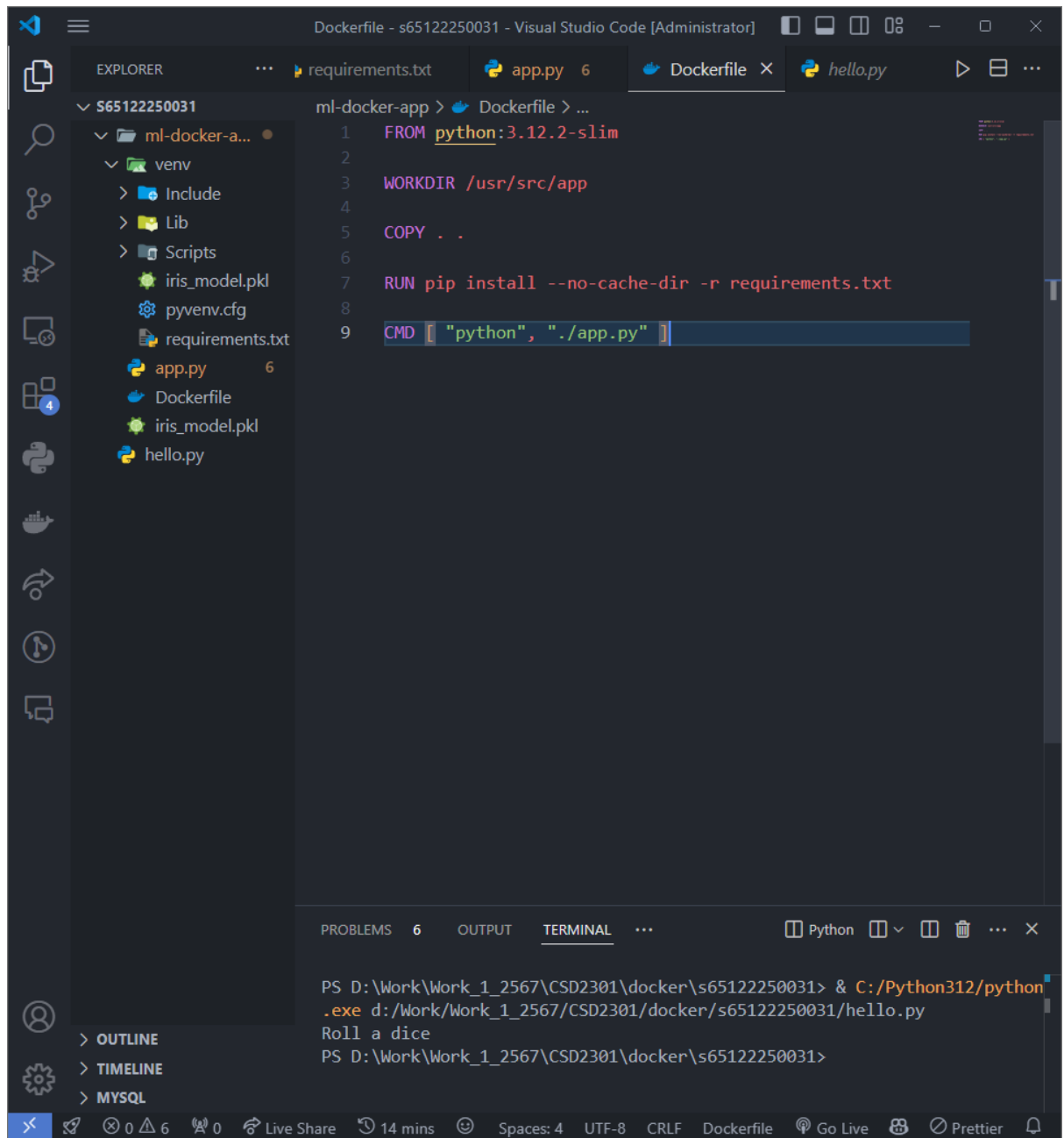
7. Install the Dependencies

```
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> pip install -r requirements.txt
Collecting scikit-learn==1.5 (from -r requirements.txt (line 1))
  Downloading scikit_learn-1.5.0-cp312-cp312-win_amd64.whl.metadata (11 kB)
Collecting pandas==2.2.2 (from -r requirements.txt (line 2))
  Downloading pandas-2.2.2-cp312-cp312-win_amd64.whl.metadata (19 kB)
Collecting joblib==1.4.2 (from -r requirements.txt (line 3))
  Downloading joblib-1.4.2-py3-none-any.whl.metadata (5.0 kB)
Collecting numpy>=1.19.5 (from scikit-learn==1.5->-r requirements.txt (line 1))
  Downloading numpy-2.1.1-cp312-cp312-win_amd64.whl.metadata (59 kB)
Collecting scipy==1.6.0 (from scikit-learn==1.5->-r requirements.txt (line 1))
  Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn==1.5->-r requirements.txt (line 1))
  Downloading threadpoolctl-3.5.0-py3-none-any.whl.metadata (13 kB)
Collecting python-dateutil<=2.8.2 (from pandas==2.2.2->-r requirements.txt (line 2))
  Downloading python_dateutil-2.9.0.post0-py3-none-any.whl.metadata (8.4 kB)
Collecting pytz==2024.1 (from pandas==2.2.2->-r requirements.txt (line 2))
  Downloading pytz-2024.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata==2024.1 (from pandas==2.2.2->-r requirements.txt (line 2))
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas==2.2.2->-r requirements.txt (line 2))
  Downloading six-1.16.0-py2.py3-none-any.whl.metadata (1.8 kB)
Download scikit_learn-1.5.0-cp312-cp312-win_amd64.whl (18.9 MB)
Download pandas-2.2.2-cp312-cp312-win_amd64.whl (11.5 MB)
Download joblib-1.4.2-py3-none-any.whl (301 kB)
Download numpy-2.1.1-cp312-cp312-win_amd64.whl (12.6 MB)
Download python_dateutil-2.9.0.post0-py3-none-any.whl (229 kB)
Download pytz-2024.1-py2.py3-none-any.whl (505 kB)
Download scipy-1.14.1-cp312-cp312-win_amd64.whl (44.5 MB)
Download threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Download tzdata-2024.1-py2.py3-none-any.whl (305 kB)
Download six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, tzdata, threadpoolctl, six, numpy, joblib, scipy, python-dateutil, scikit-learn, pandas
Successfully installed joblib-1.4.2 numpy-2.1.1 pandas-2.2.2 python-dateutil-2.9.0.post0 pytz-2024.1 scikit-learn-1.5.0 scipy-1.14.1 six-1.16.0 threadpoolctl-3.5.0 tzdata-2024.1
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv>
```

8. Run application

```
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> python app.py
Accuracy: 1.0
Model saved!
```

9. Dockerfile



The screenshot shows the Visual Studio Code interface with a Dockerfile being edited. The Explorer sidebar on the left shows a project structure for 'S65122250031' containing a 'venv' directory and files like 'iris_model.pkl', 'pyvenv.cfg', 'requirements.txt', 'app.py', 'Dockerfile', and 'hello.py'. The Dockerfile editor shows the following content:

```
1 FROM python:3.12.2-slim
2
3 WORKDIR /usr/src/app
4
5 COPY . .
6
7 RUN pip install --no-cache-dir -r requirements.txt
8
9 CMD ["python", "./app.py"]
```

The terminal at the bottom shows the command to run the Docker container and its output:

```
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031> & C:/Python312/python
.exe d:/Work/Work_1_2567/CSD2301/docker/s65122250031/hello.py
Roll a dice
PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031>
```

The status bar at the bottom indicates the file is a Dockerfile and shows various icons for Live Share, workspace settings, and other features.

10. Docker Image

```
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker build -f Dockerfile -t my-ml-app .
[+] Building 36.5s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> == transferring dockerfile: 184B                             0.0s
=> [internal] load metadata for docker.io/library/python:3.12.2-slim 1.0s
=> [internal] load .dockerignore                                0.0s
=> == transferring context: 2B                                   0.0s
=> [1/4] FROM docker.io/library/python:3.12.2-slim@sha256:5dc6f84b5e97bf0c90abfb7c55f3cacc2cb6687c8f920b64a833a 0.0s
=> [internal] load build context                                9.9s
=> == transferring context: 341.25MB                             9.9s
=> CACHED [2/4] WORKDIR /usr/src/app                           0.0s
=> [3/4] COPY . .                                              2.4s
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt    20.9s
=> exporting to image                                           2.2s
=> == exporting layers                                           2.1s
=> == writing image sha256:17a33b4d3362d4df1e0ed976f4cacc458b153fd105005c9656e81526017b2b93 0.0s
=> == naming to docker.io/library/my-ml-app                     0.0s
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker build -t ml-docker-app .
[+] Building 1.6s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> == transferring dockerfile: 184B                             0.0s
=> [internal] load metadata for docker.io/library/python:3.12.2-slim 1.0s
=> [internal] load .dockerignore                                0.0s
=> == transferring context: 2B                                   0.0s
=> [1/4] FROM docker.io/library/python:3.12.2-slim@sha256:5dc6f84b5e97bf0c90abfb7c55f3cacc2cb6687c8f920b64a833a 0.0s
=> [internal] load build context                                0.5s
=> == transferring context: 947.84kB                             0.5s
=> CACHED [2/4] WORKDIR /usr/src/app                           0.0s
=> CACHED [3/4] COPY . .                                        0.0s
=> CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt 0.0s
=> exporting to image                                           0.0s
=> == exporting layers                                           0.0s
=> == writing image sha256:17a33b4d3362d4df1e0ed976f4cacc458b153fd105005c9656e81526017b2b93 0.0s
=> == naming to docker.io/library/ml-docker-app                 0.0s
```

11. Run Docker container

```
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker image

Usage:  docker image COMMAND

Manage images

Commands:
  build      Build an image from a Dockerfile
  history    Show the history of an image
  import     Import the contents from a tarball to create a filesystem image
  inspect    Display detailed information on one or more images
  load       Load an image from a tar archive or STDIN
  ls         List images
  prune      Remove unused images
  pull       Download an image from a registry
  push       Upload an image to a registry
  rm         Remove one or more images
  save       Save one or more images to a tar archive (streamed to STDOUT by default)
  tag        Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker run -it --rm my-ml-app:latest
Accuracy: 1.0
Model saved!
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> |
```

12. Tag & Push the container to dockerhub

```
Administrator: Windows PowerShell
Administrator: Command Prompt

(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker login
Authenticating with existing credentials...
Login Succeeded
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker tag ml-docker-app mrwinrock/ml-docker-app
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> docker push mrwinrock/ml-docker-app
Using default tag: latest
The push refers to repository [docker.io/mrwinrock/ml-docker-app]
458ab882bf3f: Pushed
e2e280ef4a60: Pushed
2f751009752e: Pushed
4bbb80f01293: Mounted from library/python
1c8df5403ed1: Mounted from library/python
315f904f4c8d: Mounted from library/python
c8f253aef560: Mounted from library/python
a483da8ab3e9: Mounted from library/python
latest: digest: sha256:0cb491688ce99d4be4c4d9cbd1352691b11f157a8758b5373c33c07e5f4200e4 size: 2003
(venv) PS D:\Work\Work_1_2567\CSD2301\docker\s65122250031\ml-docker-app\venv> |
```

